

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/981,453	10/18/2001	Jens B. Junkermann	10022/093	6887

27879 7590 09/22/2004  
INDIANAPOLIS OFFICE 27879  
BRINKS HOFER GILSON & LIONE  
ONE INDIANA SQUARE, SUITE 1600  
INDIANAPOLIS, IN 46204-2033

EXAMINER

STEVENS, ROBERT

ART UNIT

PAPER NUMBER

2176

DATE MAILED: 09/22/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

# Office Action Summary

Application No.

09/981,453

Applicant(s)

JUNKERMANN, JENS B.

Examiner

Robert M Stevens

Art Unit

2176

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☒ Responsive to communication(s) filed on 18 October 2001.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1-50 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-50 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 18 October 2001 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

## Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date 10/18/01, 7/7/03.
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_\_.

### **DETAILED ACTION**

1. Claims 1-50 are pending in Application No. 09/981,453, entitled "XML-Based Multi-Format Business Services Design Pattern", filed 10/18/2001 by Junkermann.

Claims 1, 13, 21, 34 and 41 are independent.

2. The Office acknowledges two Information Disclosure Statements filed on 10/18/2001 and 7/7/2003.

3. The Office acknowledges claim to benefit of provisional priority of 60/268,981 filed 2/15/2001.

### ***Drawings***

4. The drawings are objected to because:

Fig. 9 depicts #172/174/176/178 as entities existing outside of #48, whereas the specification indicates that these elements are internal to #48. Also, the specification indicates that #178 is internal to the Backend Systems layer #18, but Fig. 9 depicts #178 as a standalone element. Please explain or correct, as discussed below.

5. Corrected drawing sheets are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should

include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The figure or figure number of an amended drawing should not be labeled as "amended." If a drawing figure is to be canceled, the appropriate figure must be removed from the replacement sheet, and where necessary, the remaining figures must be renumbered and appropriate changes made to the brief description of the several views of the drawings for consistency. Additional replacement sheets may be necessary to show the renumbering of the remaining figures. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

---

### ***Specification***

6. The disclosure is objected to because of the following informalities:
  - A. Page 15 lines 26-27: "application that use" and "provide standardize access". Please correct all spelling/grammatical/etc. mistakes throughout the specification;
  - B. Page 7 lines 6-21: This passage is confusing. It's unclear whether the embodiment being described is required to do anything ("may be", "may include", ...). For enablement/best mode purposes, description of an

exemplary embodiment should affirmatively state implementation details.

Please clarify.

Appropriate correction is required.

***Claim Rejections - 35 USC § 112***

7. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

8. **Claims 1-50 rejected under 35 U.S.C. 112, second paragraph**, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

**Regarding claims 1 (line 6), 10 (line 2), 13 (line 7), 21 (lines 9, 10, 15, and 17), 29 (line 1),** “as a function of” is indefinite language, rendering the scope of these claims indeterminable.

**Claims 2-12, 14-20 and 22-33** are dependent upon claims 1, 13 and 21, as appropriate, and therefore likewise rejected.

**Regarding claim 34**, the scope is indeterminable. This claim is directed to software comprising hardware ("software architecture comprising ... a server").

**Claims 35-40** are dependent upon claim 34, and therefore likewise rejected.

**Regarding claim 41**, the scope is indeterminable. This claim is directed to an abstract concept (business services layer) comprised of hardware and software.

**Claims 32-50** are dependent upon claim 34, and therefore likewise rejected.

**Regarding claim 2**, the scope is indeterminable. A dependent claim cannot delete/replace a parent's limitation. (Claim limitation 1a = "receive a request", but claim 2 = "generate a request")

---

**Regarding claim 8**, the scope is indeterminable. A dependent claim cannot delete/replace a parent's limitation. (Claim limitation 1d = "receive data", but claim 8 = "create a data structure")

***Claim Rejections - 35 USC § 102***

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

10. **Claims 1-6, 8-25 and 27-40 are rejected under 35 U.S.C. 102(e)** as being anticipated by Jamtgaard et al. (US Patent No. 6,430,624, provisionally filed Oct. 21, 1999 and issued Aug. 6, 2002, hereafter referred to as "Jamtgaard").

---

**Regarding independent method claim 1, Jamtgaard discloses:**

*A method of interfacing a front-end systems layer with a back-end systems layer using a self describing data structure, the method comprising:*

*a) receiving a request from a front-end systems layer; (col. 4 lines 10-12)*

*b) translating the request; (col. 4 lines 58-62)*

*c) executing custom application code to access data within a back-end systems layer as a function of the translated request; (Fig. 5 #62)*

*d) receiving data in response to the translated request from the custom application code; (Fig. 5 #62 to #44, re: "Presentation Shoe", then to #60, then #15) and*

*e) translating the data to a format defined in the request. (col. 4 line 66 – col. 5 line 3)*



**Regarding claim 2**, which is dependent upon claim 1, Jamtgaard discloses:

*wherein a) comprises generating the request in a servlet request format. (Fig. 5 connection between #60 and #44)*

**Regarding claim 3**, which is dependent upon claim 1, Jamtgaard discloses:

*wherein b) comprises translating the request to extensible markup language. (Fig. 5 connection between #60 and #62, re: "XML Document")*

**Regarding claim 4**, which is dependent upon claim 1, Jamtgaard discloses:

*wherein b) comprises translating the request into a document object model document to represent an input message. (Fig. 3 #28, re: DOM format)*

**Regarding claim 5**, which is dependent upon claim 1, Jamtgaard discloses:

*wherein b) comprises limiting the translated request to representation as at least one of integer, long, Boolean, string and group fields. (Fig. 9A, re: "group")*

**Regarding claim 6**, which is dependent upon claim 1, Jamtgaard discloses:

*wherein b) comprises generating a plurality of fields as a function of tags provided in the request. (Fig. 9A, re: group, atomic)*

**Regarding claim 8**, which is dependent upon claim 1, Jamtgaard discloses:

*wherein d) comprises establishing a structure for the response in extensible mark up language. (col. 4 lines 12-17)*

**Regarding claim 9**, which is dependent upon claim 1, Jamtgaard discloses:

*wherein d) comprises limiting the data to representation as at least one of integer, long, Boolean, string and group. (Fig. 9A, re: "group")*

**Regarding claim 10**, which is dependent upon claim 1, Jamtgaard discloses:

*wherein d) comprises generating a document object model document to represent an output message as a function of data received. (col. 10 lines 51-60)*

**Regarding claim 11**, which is dependent upon claim 1, Jamtgaard discloses:

*wherein d) comprises generating a document object model document to represent an output message as a function of data received. (col. 4 line 66 – col. 5 line 3)*

---

**Regarding claim 12**, which is dependent upon claim 1, Jamtgaard discloses:

*further comprising f) returning the translated data to the front-end systems layer. (Fig. 5 #62, 44)*

**Regarding independent method claim 13**, Jamtgaard discloses:

*A method of leveraging extensible markup language technology to interface a front-end systems layer with a back-end systems layer, the method comprising:*

*a) receiving a request initiated with a delivery technology; (col. 4 lines 10-12)*

*b) identifying the value of a request name parameter from the request; (col. 6 lines 60-66)*

*c) translating the request to an input message, the input message comprising a root element and a plurality of sub elements; (Fig. 5 input connection from #60 to #62, re: "XML Document") and*

Art Unit: 2176

*d) initiating the retrieval of data as a function of the request name parameter. (Fig. 5 #62 to #44, re: "Presentation Shoe", then to #60, then #15)*

**Regarding claim 14**, which is dependent upon claim 13, Jamtgaard discloses:

*further comprising:*

*e) providing data as a response; (Fig. 5 #62 to #44, re: "Presentation Shoe", then to #60, then #15)*

*f) creating an output message with the response, the output message comprising a root element and a plurality of sub-elements; (col. 5 lines 54-64) and*

*g) translating the output message to a format compatible with the delivery technology. (col. 4 line 66 – col. 5line 6)*

**Regarding claim 15**, which is dependent upon claim 13, Jamtgaard discloses:

*wherein c) comprises setting the root element to a message name as a function of the request name parameter. (Fig. 17 #300, as discussed at col. 17 lines 43-45)*

**Regarding claim 16**, which is dependent upon claim 13, Jamtgaard discloses:

*wherein c) comprises creating a document object model document. (Fig. 3 #28, re: DOM format)*

**Regarding claim 17**, which is dependent upon claim 13, Jamtgaard discloses:

*wherein d) comprises executing custom application code corresponding to the request name parameter. (Fig. 5 #62)*

**Regarding claim 18**, which is dependent upon claim 14, Jamtgaard discloses:

*wherein 1) comprises creating a document object model document. (Fig. 3 #28, re: DOM format)*

**Regarding claim 19**, which is dependent upon claim 14, Jamtgaard discloses:

*wherein f) comprises setting the root element to a message name as a function of the request name parameter. (Fig. 17 #300, as discussed at col. 17 lines 43-45)*

**Regarding claim 20**, which is dependent upon claim 13, Jamtgaard discloses:

*wherein the delivery technology comprises at least one of an Internet browser, a telephone, a wireline communication device, a wireless communication device and a wireless application protocol device. (col. 4 lines 58-66)*

**Regarding independent method claim 21**, Jamtgaard discloses:

*A method of operating a business services application for retrieving data with delivery technologies, the method comprising:*

- a) developing custom application code (Fig. 5 #62) in a subclass of a BusinessService class (Fig. 5, un-numbered dotted region), the custom application code responsive to a request for data initiated by the delivery technologies; (Fig. 5 #62, 44, 60 and 15)*
- b) translating the request to a first document object model document with an ApiService class; (Fig. 3 #28, re: DOM format)*
- c) selectively limiting the data structure of the first document object model document as a function of a Message class and a Field class; (tree data structure depicted in Fig. 14)*
- d) executing the custom application code to retrieve data as a function of the first document object model document; (Fig. 5 connection from #44 to #62, re: "Device info" and "Shoe ID")*
- e) reading data into a second document object model document with the ApiService class; (col. 9 lines 51-60)*
- f) selectively limiting the data structure of the second document object model document as a function of the Message class and the Field class; (tree data structure depicted in Fig. 14) and*
- g) translating the second document object model document with the ApiService class as a function of the delivery technology. (col. 4 line 66 – col. 5 line 3)*

**Regarding claim 22**, which is dependent upon claim 21, Jamtgaard discloses:

*wherein c) comprises setting a plurality of text nodes within the first document object model document to a unit of data identified by a tag in the request. (Fig. 9A #122, 124, 126)*

**Regarding claim 23**, which is dependent upon claim 22, Jamtgaard discloses:

*wherein c) further comprises limiting the unit of data to a predetermined data type. (Fig. 9A #122, where name = data type String [e.g., "News"])*

**Regarding claim 24**, which is dependent upon claim 22, Jamtgaard discloses:

*wherein c) further comprises limiting the predetermined data type to a string. (Fig. 9A #122, where name = data type String [e.g., "News"])*

**Regarding claim 25**, which is dependent upon claim 21, Jamtgaard discloses:

*wherein c) comprises setting an attribute node within the first document object model document to an attribute identified by a request name parameter in the request. (Fig. 17 #300, as discussed at col. 17 lines 43-45)*

**Regarding claim 27**, which is dependent upon claim 21, Jamtgaard discloses:

*wherein b) comprises representing an input message with the first document object model document. (Fig. 3 #28, re: DOM format)*

**Regarding claim 28**, which is dependent upon claim 21, Jamtgaard discloses:

*wherein e) comprises representing an output message with the second document object model document. (Fig. 3 #28, re: DOM format)*

**Regarding claim 29**, which is dependent upon claim 21, Jamtgaard discloses:

*wherein f) comprises setting, as a function of a data type, a plurality of text nodes within the second document object model document to data*

*read in to the second document object model document. (Fig. 17 and col. 17 lines 43-53)*

**Regarding claim 30**, which is dependent upon claim 21, Jamtgaard discloses:

*wherein f) comprises setting, as a function of a data type, an attribute node within the second document object model document to all attribute read in to the second document object model document with the data. (Fig. 17 and col. 17 lines 43-53)*

**Regarding claim 31**, which is dependent upon claim 30, Jamtgaard discloses:

*wherein the attribute comprises an attribute name and an attribute value and f) further comprises limiting the attribute value to a predetermined data type. (Fig. 9A #122, where name = data type String [e.g., "News"])*

**Regarding claim 32**, which is dependent upon claim 21, Jamtgaard discloses:

*wherein g) comprises translating the second document object model document to extensible markup language text. (col. 4 line 66 – col. 5 line 3)*

**Regarding claim 33**, which is dependent upon claim 21, Jamtgaard discloses:

*wherein g) comprises translating the second document object model document to at least one of a hypertext markup language and a website meta language as a function of at least one extensible stylesheet language stylesheet. (col. 11 lines 9-12)*

**Regarding independent claim 34**, Jamtgaard discloses:

*An e-commerce software architecture for providing a framework to interface delivery technologies with data, the e-commerce software architecture comprising:*

*a server computer operable to execute instructions to convert a request to an input message in a predetermined extensible markup language format, the input message comprising a plurality of request parameters; (col. 6 lines 34-40 and col. 4 line 66 – col. 5 line 3)*

*the server computer operable to execute instructions to retrieve data as a function of the request parameters, (col. 6 lines 49-53, alluding to cookies as parameters)*

*the server computer operable to execute instructions to create an output message in a predetermined extensible markup language format, the output message comprising the data retrieved, (col. 4 line 66 – col.5 line 3) and*

*the server computer operable to execute instructions to convert the output message to a format indicated by the request. (col. 4 line 66 – col.5 line 3)*

**Regarding claim 35**, which is dependent upon claim 34, Jamtgaard discloses:

*wherein the request comprises a servlet request format. (Fig. 5 connection between #60 and #44)*

**Regarding claim 36**, which is dependent upon claim 34, Jamtgaard discloses:

*wherein the predetermined extensible markup language format of the input message and the output message is predetermined as a function of instructions comprising a createField method and a setAttribute method. (Fig. 5 connection from #60 to #62, re: "XML Document" and "Presentation Shoe")*

**Regarding claim 37**, which is dependent upon claim 34, Jamtgaard discloses:

*wherein the instructions to convert a request to an input message comprises a createInputMessage method, a createField method and a setAttribute method. (col. 6 lines 34-40 and col. 4 line 66 – col. 5 line 3)*

**Regarding claim 38**, which is dependent upon claim 34, Jamtgaard discloses:

*wherein the instructions to create an output message comprises a createOutputMessage method, a createField method and a setAttribute method. (col. 4 line 66 – col. 5 line 3)*

**Regarding claim 39**, which is dependent upon claim 34, Jamtgaard discloses:

*wherein the instructions to retrieve data comprises custom application code. (Fig. 5 #62)*

**Regarding claim 40**, which is dependent upon claim 34, Jamtgaard discloses:

*wherein the instructions to convert the output message comprises one of a generateXML- method and a generatePresentation method. (col. 4 line 66 – col. 5 line 6, discussing XML generation and presentment)*

### ***Claim Rejections - 35 USC § 103***

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

12. **Claims 7 and 26 are rejected under 35 U.S.C. 103(a)** as being unpatentable over Jamtgaard et al. (US Patent No. 6,430,624, provisionally filed Oct. 21, 1999 and issued Aug. 6, 2002, hereafter referred to as “Jamtgaard”) in view of Chuck McManis, “Take an in-depth look at the Java Reflection API”, Java World, September 1997, pp. 1-11, hereafter referred to as “McManis”).

**Regarding claim 7**, which is dependent upon claim 1, the limitations of claim 1 have been previously discussed.



However, Jamtgaard does not explicitly disclose:

*wherein b) comprises selectively setting the length of a field name for each of a plurality of fields.*

McManis, though, discloses:

*wherein b) comprises selectively setting the length of a field name for each of a plurality of fields. (p. 4 code under section entitled "Identifying the class's package", especially the code referring to "y = x.substring ..." and "y.length()")*

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of McManis for the benefit of Jamtgaard, because to do so would allow a programmer to identify a class's package as taught by McManis in p. 4, "Identifying the class's package" section. These references were all applicable to the same field of endeavor, i.e., object oriented programming.

---

**Regarding claim 26**, which is dependent upon claim 21, the limitations of claim 21 have been previously discussed.

However, Jamtgaard does not explicitly disclose:

*further comprising selecting, as a function of a mode debug flag, one of a short field name and a long field name for each of a plurality of fields in the first and second document object model documents.*

McManis, though, discloses:

*further comprising selecting, as a function of a mode debug flag, one of a short field name and a long field name for each of a plurality of fields in the first and second document object model documents. (p. 5, 2<sup>nd</sup> paragraph under section entitled "Collecting class references from declarations and parameters")*

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of McManis for the benefit of Jamtgaard, because to do so would allow a programmer to identify a shorthand name for a type as taught by McManis in p. 5, 2<sup>nd</sup> paragraph under section entitled "Collecting class references from declarations and parameters". These references were all applicable to the same field of endeavor, i.e., object oriented programming.

13. **Claims 41-50 are rejected under 35 U.S.C. 103(a)** as being unpatentable over Jamtgaard et al. (US Patent No. 6,430,624, provisionally filed Oct. 21, 1999 and issued Aug. 6, 2002, hereafter referred to as "Jamtgaard") in view of David Flanagan, Java Examples in a Nutshell: A Tutorial Companion to Java in a Nutshell, O'Reilly & Associates, Inc., (c) 1997, pp. 20-26, hereafter referred to as "Flanagan"). Note that the Microsoft Dictionary, 5<sup>th</sup> Edition, Microsoft Press, Redmond, © 2002 (hereafter "MS Dictionary") has been used to provide the definition of "wrapper".

**Regarding independent method claim 41, Jamtgaard discloses:**

*A business services layer for leveraging extensible markup language technology to provide an interface between a back-end systems layer and a front-end systems layer, the business services layer comprising:*  
*a server computer; (col. 4 lines 39-40)*  
*an ApiService class operable within the server computer to direct the translation of a request to an input message; (Fig. 5 #44)*

Art Unit: 2176

*a document object model class operable within the server computer to represent the input message as a document object model document; (Fig. 3 #28, re: use of DOM)*  
*and*  
*a BusinessService class operable within the server computer to direct the execution of custom application code as a function of the input message. (Fig. 5, un-numbered dotted region [encloses #62-84])*

However, Jamtgaard does not explicitly disclose:

*a Message class and a Field class operable within the server computer as wrapper of the document object model class to restrict manipulation of the document object model document;*

Flanagan, though, discloses:

*a Message class and a Field class operable within the server computer as wrapper of the document object model class to restrict manipulation of the document object model document; (p. 24, 1<sup>st</sup> paragraph under "Complex Numbers" discusses encapsulation. Note that MS Dictionary [p. 575] defines a wrapper as an object which "encapsulates" another object)*

---

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Flanagan for the benefit of Jamtgaard, because to do so would allow a programmer to hide implementation details as taught by Flanagan in p. 24, 1<sup>st</sup> paragraph under "Complex Numbers". These references were all applicable to the same field of endeavor, i.e., object oriented programming.

**Regarding claim 42**, which is dependent upon claim 41, the limitations of claim 41 have been previously discussed. Jamtgaard further discloses:

*wherein the custom application code is operable to process the input message to retrieve data, the data translatable with the document object model class, the Message class and the Field class to an output message in the form of a document object model document. (Fig. 5 #62)*

**Regarding claim 43**, which is dependent upon claim 42, the limitations of claim 41 have been previously discussed. Jamtgaard further discloses:

*wherein the ApiService class is operable to direct the conversion of the output message to a presentation format defined by the request. (col. 4 line 66 – col. 5 line 6)*

**Regarding claim 44**, which is dependent upon claim 41, the limitations of claim 41 have been previously discussed. Jamtgaard further discloses:

*wherein the input message and the output message comprises a root element and a plurality of sub-elements. (Fig. 12, showing a tree data structure comprised of a root node and sub-element nodes)*

**Regarding claim 45**, which is dependent upon claim 41, the limitations of claim 41 have been previously discussed. Jamtgaard further discloses:

*further comprising a Fldtypes class operable within the server computer, wherein the Fldtypes class comprises definitions of the format of datatypes for fields within the input message. (col. 14 lines 50-60)*

**Regarding claim 46**, which is dependent upon claim 41, the limitations of claim 41 have been previously discussed. Jamtgaard further discloses:

*wherein the document object model document comprises a plurality of field names, the field names selectable with a mode debug flag as one of a first field name and a second field name. (Fig. 3 #28, re: use of DOM)*

**Regarding claim 47**, which is dependent upon claim 46, the limitations of claim 41 have been previously discussed. Jamtgaard further discloses:

*wherein the first field name and the second field name are defined in a MESSAGEDEFINITION class operable within the server computer. (col. 12 lines 53-63 discuss field and atomic names)*

**Regarding claim 48**, which is dependent upon claim 41, the limitations of claim 41 have been previously discussed. Jamtgaard further discloses:

*wherein the document object model class comprises a Document class, a document object model Element class and a plurality of ProcessingInstruction classes, the Message class operable as a wrapper of the Document class, the document object model Element class and the ProcessingInstruction classes. (Fig. 3 #28, re: use of DOM)*

---

**Regarding claim 49**, which is dependent upon claim 41, the limitations of claim 41 have been previously discussed. Jamtgaard further discloses:

*wherein the document object model class comprises a document object model setAttribute method, Field class operable as a wrapper of the document object model setAttribute method. (Fig. 3 #28, re: use of DOM)*

**Regarding claim 50**, which is dependent upon claim 41, the limitations of claim 41 have been previously discussed.

However, Jamtgaard does not explicitly disclose:

*wherein the BusinessService class comprises a subclass of custom application code responsive to the request.*

Flanagan, though, discloses:

*wherein the BusinessService class comprises a subclass of custom application code responsive to the request. (p. 23 "A Rect Subclass" section discusses the use of a subclass)*

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Flanagan for the benefit of Jamtgaard, because to do so would allow a programmer to create a child class that inherits the fields and methods of its parent class as taught by Flanagan in p. 23 "A Rect Subclass" section. These references were all applicable to the same field of endeavor, i.e., object oriented programming.

### **Conclusion**

14. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

### **Non-patent Literature**

McCluskey, Glen, "Using Java Reflection", The Source for Developers: A Sun developer Network Site, January 1998, downloaded from: [java.sun.com/developer/technicalArticles/ALT/Reflection/](http://java.sun.com/developer/technicalArticles/ALT/Reflection/), pp. 1-7.

### **US Patent Application Publications**

Capers et al	US2002/0112055
Lipkin et al	US2002/0120859

### **US Patents**

Schwerdtfeger et al	6,725,424
Britton et al	6,535,896
Burkett et al	6,635,089
Fernandez et al	6,604,100
Wanderski et al	6,519,617

Allen	6,658,625
Yalcinalp	6,507,857
Sundaresan	6,487,566
Lau	6,598,219

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Robert M Stevens whose telephone number is (703) 605-4367. The examiner can normally be reached on M-F 7:00 - 3:30. After mid-October 2004, the Examiner can be reached at (571) 272-4102.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Joseph Feild can currently be reached on (703) 305-9792. The current fax phone number for the organization where this application or proceeding is assigned is 703-872-9306. However, note that the main number for Technology Center 2100 will be (571) 272-2100, as of mid-October 2004.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Robert M. Stevens  
Art Unit 2176  
Date: September 14, 2004

  
**JOSEPH FEILD**  
**SUPERVISORY PATENT EXAMINER**

rms